

# **Optimalizace webových portálů DB Education a DB Manager**

Optimization of Web Portals DB Education and DB Manager

**Aleš Děcký**

Bakalářská práce

Vedoucí práce: Ing. Peter Chovanec, Ph.D.

Ostrava, 2021

## **Abstrakt**

Motivem této bakalářské práce je optimalizace informačních systémů DB Education a DB Manager. Cílem bylo prozkoumat fungování původních verzí a navrhnout tak řešení jejich nedostatků. Práce vyobrazuje popis původních systémů, zejména jejich funkční a datovou analýzu. V rámci popisu systému je obsažena také i deskripce datové vrstvy zahrnující charakteristiku databáze, její komunikace a práce s touto databází v kódu aplikace. Optimalizované verze systémů pak reflektují tyto popisy a vyobrazují provedené změny. Výsledek práce zahrnuje i popis přenosu původních dat systémů do databází optimalizovaných systémů.

**Klíčová slova:** Informační systém, Databázové servery, Výukový portál, DB Education, DB Manager, Microsoft SQL Server, Entity Framework Core

## **Abstract**

The motive of the bachelor thesis is the optimization of information systems DB Education and DB Manager. The aim was to examine the functioning of the original version and propose solutions to their shortcomings. The thesis shows a description of the original systems, especially their functional and data analysis. The description of the system also includes explanation of the data layer, including the characteristics of the database, its communication and work with this database in the application code. The optimized versions of the systems then reflect on these descriptions and illustrate the changes made. The result of the bachelor thesis also includes a description of the transfer of the original system data to the databases of optimized systems.

**Key Words:** Information system, Database servers, Educational portal, DB Education, DB Manager, Microsoft SQL Server, Entity Framework Core

Rád bych chtěl tímto poděkovat svému vedoucímu práce Ing. Peterovi Chovancovi, Ph.D., za ochotu a podporu během vypracovávání této bakalářské práce. Dále bych chtěl taky poděkovat Ing. Lukáši Zátopkovi za ochotu a vysvětlení frameworku Entity Framework Core a dalších možností při vývoji aplikací v .NET a ASP.NET Core. V neposlední řadě bych chtěl poděkovat i mým přátelům, rodině a přítelkyni, kteří mě celou dobu podporovali.

## Obsah

Seznam použitých symbolů a zkratk.....	5
Seznam obrázků.....	6
Seznam tabulek .....	7
Seznam výpisů zdrojového kódu .....	8
1 Úvod .....	9
2 DB Education 1.0 .....	10
2.1 O verzi.....	10
2.2 Funkční analýza .....	10
2.3 Datová analýza .....	15
2.4 Datová vrstva.....	17
3 DB Education 2.0 .....	18
3.1 O verzi.....	18
3.2 Funkční analýza .....	18
3.3 Datová analýza .....	21
3.4 Datová vrstva.....	22
4 DB Manager 1.0.....	23
4.1 O verzi.....	23
4.2 Funkční analýza .....	23
4.3 Datová analýza .....	27
4.4 Datová vrstva.....	28
5 DB Manager 2.0.....	30
5.1 O verzi.....	30
5.2 Funkční analýza .....	30
5.3 Datová analýza .....	33
5.4 Datová vrstva.....	34
6 Konvertor dat .....	36
7 Aplikační a prezentační vrstva.....	37
8 Závěr .....	38
Literatura .....	39
A Elektronická příloha.....	41

## Seznam použitých symbolů a zkratk

API	- Application programming interface
CTE	- Common table expressions
HTML	- Hypertext markup language
JSON	- JavaScript object notation
LDAP	- Lightweight directory access protocol
LINQ	- Language integrated query
MVC	- Model view controller
ORM	- Object-relational mapping
POCO	- Plain old CLR object
SMTP	- Simple mail transfer protocol
SQL	- Structured query language
URL	- Uniform resource locator
XML	- Extensible markup language

## Seznam obrázků

Obrázek 1: Relační schéma databáze DB Education 1.0 .....	16
Obrázek 2: Relační schéma databáze DB Education 2.0 .....	22
Obrázek 3: Relační schéma databáze DB Manager 1.0 .....	28
Obrázek 4: Relační schéma databáze DB Manager 2.0 .....	34

## Seznam tabulek

Tabulka 1: Odebrané sekce z DB Education 2.0 .....	18
Tabulka 2: Upravené sekce DB Education 2.0 .....	18
Tabulka 3: Změněné funkce DB Education 2.0 .....	19
Tabulka 4: Metody rozhraní IConnections .....	29
Tabulka 5: Upravené sekce DB Manager 2.0.....	30
Tabulka 6: Změněné funkce DB Manager 2.0.....	31
Tabulka 7: Optimalizované metody DB Managera 2.0.....	35

## **Seznam výpisů zdrojového kódu**

Výpis 1: Ukázka převodu dat DB Education.....	37
---	----



# 1 Úvod

Vývoj informačního systému je často složitý a dlouho trvající proces. Na začátku je velmi složité definovat jeho požadavky. Vývoj angažuje několik lidí a výsledný systém není tak dokonalý, jak byl původně zamýšlen. Objevují se problémy, které je potřeba optimalizovat. S některými problémy se potýkají také informační systémy DB Education a DB Manager.

DB Education je webový informační systém určený pro podporu výuky databázových předmětů, který poskytuje studentům studijní materiály z přednášek a cvičení, instalační soubory, skripta, různá videa, manuály a podobně. Součástí systému je také odevzdávání projektů v rámci vyučovaných předmětů.

DB Manager je informační systém pro správu databázových systémů Oracle [1], Microsoft SQL Server [2], MySQL [3], PostgreSQL [4] využívaných při výuce databázových předmětů, a taktéž při vypracovávání bakalářských, diplomových a disertačních prací.

Uživatelé obou systémů může být pouze pracovník anebo student VŠB-Technická Univerzita Ostrava, protože vstupní autorizace využívá univerzitní LDAP [5].

Cílem bakalářské práce bude optimalizace těchto webových informačních systémů. Oba systémy byly v minulých letech vypracovány nebo částečně optimalizovány jako bakalářské nebo diplomové práce. Došlo k jejich úspěšnému nasazení a jsou využity při výuce databázových systémů na fakultě elektrotechniky a informatiky Vysoké školy báňské – Technické univerzity Ostrava. Přesto však tyto naimplementované systémy vykazují nedostatky.

Pro jednoznačné definování nedostatků, dojde k provedení důkladné analýzy původních systémů. Analýza bude obsahovat funkční a datovou analýzu, včetně analýzy vrstvy k přístupu dat. Výsledkem analýzy bude přehledný popis provedených změn a implementačních kroků vedoucí k odstranění nalezených problémů.

Při implementaci optimalizovaných webových informačních systémů, dojde ke sjednocení implementačních postupů a využitých moderních technologií.

## 2 DB Education 1.0

### 2.1 O verzi

Informační systém DB Education 1.0 byl v roce 2018 vytvořen Ing. Janem Velčovským jako diplomová práce [6]. Verze systému byla vytvořena jako single-page aplikace za využití front-end frameworku Angular [7]. Ten komunikoval pomocí API s backend vrstvou implementovanou v ASP.NET Core [8]. Tato verze poskytovala poměrně rozsáhlý logovací systém, který logoval jednotlivé události v systému. Administrátoři měli k dispozici správu uživatelských rolí s možností definovat jejich působnost v systému. Tím, že verze byla implementovaná jako single-page aplikace, prezentační vrstva byla poměrně interaktivní, ovšem za cenu prvotního načtení celé javascriptové knihovny se všemi nutnými závislostmi. nabízí uživatelům příjemné uživatelské prostředí se samotnou navigací v systému.

### 2.2 Funkční analýza

Se systémem pracují tři typy uživatelů: Administrátor, Učitel a Student. Systém bude mít taktéž svou vlastní roli.

- *Administrátor*  
Stará se o chod celého systému. V jeho kompetenci je správa uživatelských rolí, správa akademických roků, předmětů, projektů a taktéž může hodnotit nebo nahlížet do všech odevzdaných projektů.
- *Učitel*  
Spravuje obsah předmětů, nahrává do systému nové studijní materiály (přednášky, cvičení...). Taktéž může hodnotit a nahlížet do projektů, u kterých ho Student specifikoval jako cvičícího.
- *Student*  
Může si prohlížet obsah předmětů a stahovat poskytované výukové materiály. Student může taktéž odevzdávat projekty a nahlížet na jejich hodnocení.
- *Systém*  
Specializovaná role, která má v kompetenci logování chodu systému nebo posílání emailů.

S neveřejnou částí webového informačního systému budou moci pracovat pouze uživatelé, již se nacházejí ve výše definovaných rolích. Veřejná část systému bude přístupna všem bez nutnosti se v roli nacházet.

### 1. Autentizace

- 1.1. Přihlášení do systému – Uživatelé jsou přihlášení do systému za použití univerzitních údajů ověřených prostřednictvím LDAP protokolu.
- 1.2. Odhlášení ze systému – Uživatelé jsou odhlášení ze systému.

## **2. Správa uživatelů**

Role: Administrátor

- 2.1. Výpis uživatelů – vrací seznam uživatelů.
- 2.2. Detail uživatele – vrací detail vybraného uživatele.
- 2.3. Přidání uživatele – přidá nový záznam uživatele.
- 2.4. Editace uživatele – upraví vybraného uživatele.
- 2.5. Změna stavu uživatele – změní stav uživatele z neaktivního na aktivní a naopak.
- 2.6. Smazání uživatele – odebere vybraného uživatele.

## **3. Správa uživatelských skupin**

Role: Administrátor

- 3.1. Výpis uživatelských skupin – vrací seznam všech uživatelských skupin.
- 3.2. Detail uživatelské skupiny – vrací detail vybrané uživatelské skupiny.
- 3.3. Přidání nové uživatelské skupiny – přidá nový záznam uživatelské skupiny.
- 3.4. Úprava uživatelské skupiny – upraví vybranou uživatelskou skupinu.
- 3.5. Odebrání uživatelské skupiny – odebere vybranou uživatelskou skupinu.
- 3.6. Výpis pravomocí uživatelské skupiny – vrací seznam všech přiřazených pravomocí uživatelské skupiny.
- 3.7. Výpis nepřiřazených pravomocí uživatelské skupiny – vrací seznam všech nepřiřazených pravomocí uživatelské skupiny.
- 3.8. Přidání pravomoci uživatelské skupiny – přidá novou pravomoci vybrané uživatelské skupině.
- 3.9. Úprava pravomoci uživatelské skupiny – upraví pravomoci vybrané uživatelské skupině.
- 3.10. Odebrání pravomoci uživatelské skupiny – odebere pravomoci vybrané uživatelské skupině.

## **4. Správa akademických roků**

Role: Administrátor, Učitel

- 4.1. Výpis záznamů akademických roků – vrací seznam všech akademických roků.
- 4.2. Detail akademického roku – vrací detail vybraného akademického roku.
- 4.3. Přidání nového akademického roku – přidá nový záznam akademického roku.
- 4.4. Editace akademického roku – upraví jméno nebo status vybraného akademického roku.
- 4.5. Smazání akademického roku – smaže vybraný akademický rok.

## **5. Správa předmětů**

Role: Administrátor, Učitel

- 5.1. Výpis předmětů – vrací seznam všech aktuálních předmětů.
- 5.2. Detail předmětu – vrací detail vybraného předmětu.
- 5.3. Přidání nového předmětu – přidá nový záznam předmětu.
- 5.4. Editace předmětu – upraví záznam vybraného předmětu.
- 5.5. Smazání předmětu – smaže vybraný předmět.

## **6. Správa kurzů**

Role: Administrátor, Učitel

- 6.1. Výpis kurzů – vrací seznam všech kurzů podle akademického roku.
- 6.2. Detail kurzu – vrací detail vybraného kurzu.
- 6.3. Přidání nového kurzu – přidá nový záznam kurzu předmětu v akademickém roce.
- 6.4. Editace kurzu – upraví záznam vybraného kurzu.
- 6.5. Smazání kurzu – smaže vybraný kurz.

## **7. Správa statických stránek**

Role: Administrátor, Učitel

- 7.1. Výpis statických stránek – vrací seznam všech statických stránek podle zvolené jazykové mutace.
- 7.2. Detail statické stránky – vrací detail vybrané statické stránky.
- 7.3. Přidání nové statické stránky – vytvoří novou statickou stránku.
- 7.4. Editace statické stránky – upraví vybranou statickou stránku.
- 7.5. Smazání statické stránky – smaže vybranou statickou stránku.

## **8. Správa projektů – specifikace**

Role: Administrátor, Učitel

- 8.1. Výpis specifikací projektů – vrací seznam všech specifikací projektů vybraného kurzu v aktuálním akademickém roce.
- 8.2. Vytvoření specifikace projektu – vytvoří novou specifikaci projektu.
- 8.3. Editace specifikace projektu – upraví vybranou specifikaci projektu.
- 8.4. Smazání specifikace projektu – smaže vybranou specifikaci projektu.

Role: Administrátor, Učitel, Student

- 8.5. Detail specifikace projektu – vrací detail vybrané specifikace projektu.

## **9. Správa projektů – řešení**

Role: Administrátor, Učitel

- 9.1. Výpis odevzdaných projektů – vrací seznam odevzdaných řešení podle akademického roku, předmětu a projektu. Je možné vyfiltrovat i jeho stav. Učitelům se zobrazí projekty přidělené pouze jim.
- 9.2. Detail projektu – zobrazí informace odevzdaného řešení obsahující mimo jiné také seznam jeho nahraných revizí.
- 9.3. Hodnocení projektu – nastaví hodnocení odevzdanému řešení a poté volá funkci 9.8.
- 9.4. Změna stavu projektu – změní stav projektu a poté volá funkci 9.8.

- 9.5. Přidání komentáře – přidá nový komentář k odevzdanému řešení a poté volá funkci 9.8. Autor komentáře má možnost nastavit komentář jen jako poznámku, jenž vidí pouze on sám.
- 9.6. Přesun projektu jinému učiteli – přesune odevzdaný projekt k jinému učiteli.
- 9.7. Přesun projektu do jiného předmětu – přesune odevzdaný projekt do jiného předmětu.

Role: Systém

- 9.8. Odesílání informačních zpráv – posílá informační zprávu studentovi prostřednictvím SMTP klienta.

Role: Administrátor, Učitel, Student

- 9.9. Výpis nahraných revizí – vrací seznam nahraných revizí s datem a časem jejich nahrání.
- 9.10. Stažení přílohy projektu – poskládá cestu uloženého souboru daného řešení a následně stáhne soubor.
- 9.11. Zobrazení komentářů – vrací seznam komentářů i poznámek k danému řešení.

Role: Student

- 9.12. Výpis hodnocení – vrací seznam odevzdaných řešení projektu daného studenta.
- 9.13. Přidání komentáře ke svému řešení – spouští funkci 9.5 bez možnosti nastavení poznámky.
- 9.14. Odevzdání řešení – vytvoří řešení projektu k vybrané specifikaci a zvaliduje typ nahrávaného souboru, který se následně nahraje na server.

## 10. Zápočty

Role: Administrátor, Učitel

- 10.1. Výpis zápočtů – vrací seznam zápočtů daného akademického roku.
- 10.2. Detail zápočtu – vrací detail vybraného zápočtu.
- 10.3. Zobrazení přihlášených uživatelů – vypisuje uživatele přihlášené ke zvolenému zápočtu.
- 10.4. Vytvoření nového zápočtu – vytvoří nový zápočet obsahující kurz, datum a čas konání, místo, kapacitu.
- 10.5. Editace zápočtu – upraví atributy vybraného zápočtu.
- 10.6. Odstranění zápočtu – odstraní vybraný zápočet ze systému.

Role: Student

- 10.7. Přihlášení na zápočet – přihlásí uživatele ke zvolenému zápočtu.

## **11. Správa online testů**

Role: Administrátor, Učitel

- 11.1. Výpis výsledků studentů – vrací seznam výsledků studentů podle vybraného testu.
- 11.2. Vytvoření nového testu – vytvoří nový záznam testu obsahující název, datum a čas spuštění testu, minimální a maximální počet bodů, heslo k zobrazení zadání testu, heslo pro zadání výsledků a zadání testu.
- 11.3. Editace testu – upraví vybraný test.
- 11.4. Smazání testu – odstraní vybraný test.
- 11.5. Hodnocení testů – nastaví danému studentovi bodový zisk v testu.

Role: Administrátor, Učitel, Student

- 11.6. Výpis testů – vrací specifikovaný seznam podle akademického roku a kurzu.

Role: Student

- 11.7. Zobrazení zadání testu – vrací zadání vybraného testu.
- 11.8. Zobrazení výsledku testu – vrací získaný počet bodů ve vybraném testu.

## **12. Logování**

Role: Administrátor

- 12.1. Výpis logů – vrací seznam logů.
- 12.2. Detail logu – zobrazí detailní informace vybraného logu.
- 12.3. Synchronizace logů – načte logy z alternativního úložiště a uloží je do systému.

Role: Systém

- 12.4. Vložení logu – vytvoří nový záznam a uloží jej do systému. V případě výpadku systému využije alternativního úložiště pro uložení záznamu.

## **13. Nezařazené**

Role: Administrátor, Učitel, Student

- 13.1. Odebírání novinek – nastaví možnost odebírání novinek vybraného kurzu.

## 2.3 Datová analýza

Systém řeší hlavně vyučované předměty, studijní materiály a studentské projekty. U každého předmětu se povinně eviduje, ve kterém akademickém roce se učí a v jakém jazyce probíhá výuka. Předmět je možné v případě potřeby schovat, případně smazat. Akademické roky je možné archivovat, přičemž systém poskytuje možnost prohlížet si archivované roky a všechny předměty v nich. Každý předmět má svůj obsah a k němu jednotlivé studijní materiály. Tento obsah je možné dynamicky upravovat. K některým vyučovaným předmětům se pojí i přihlašování k zápočtu. Zápočet nese informace o jejich názvu, času konání a čas do kdy se student může student k zápočtu přihlásit, místnost vykonání zápočtu, kapacita studentů, poznámky a jejich seznam přihlášených studentů. Krom zápočtů jsou k některým předmětům spjaty i testy. Test se definuje jeho názvem, časem zahájení a ukončení, minimálním a maximálním počtem dosažitelných bodů, loginem učitele, který test vytvořil a zadáním spolu s jeho heslem pro zobrazení, aby se k němu například nedostala jiná skupina. K testu se pojí výsledky studentů, obsahující login studenta, čas spuštění testu a jeho počet udělených bodů.

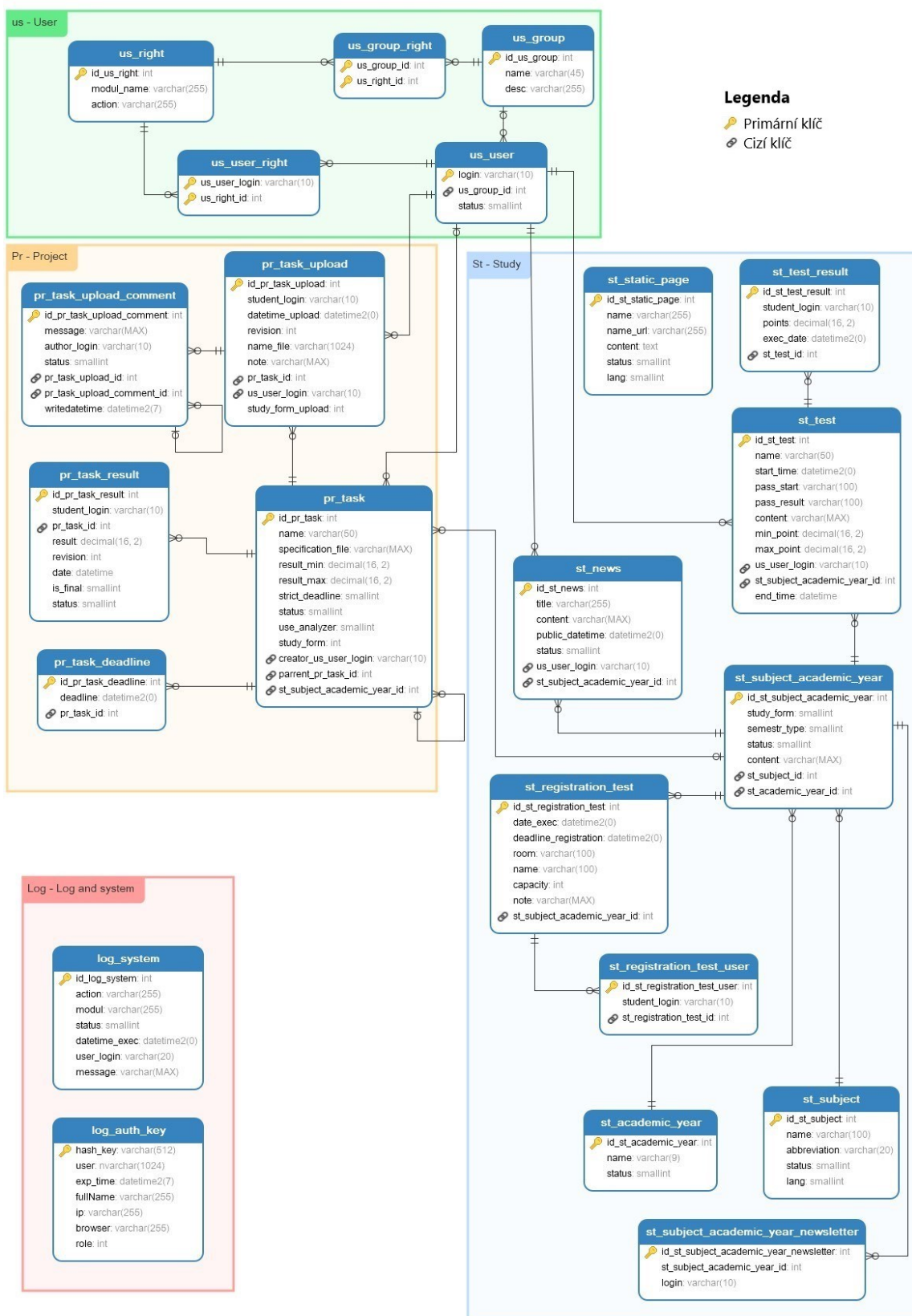
Jeden z dalších pilířů systému je správa a odevzdávání projektů, přičemž každý projekt je definovaný názvem, požadovaným formátem (koncovka souboru), minimálním a maximálním počtem dosažitelných bodů, uživatelem, který projekt vytvořil a informace, jestli je možné projekt odevzdat i po definovaném termínu. Samozřejmostí je odevzdávání složených projektů (tzn. Projektů složených z několika samostatně hodnotitelných částí). Systém taktéž pracuje s předpřipravenou knihovnou na automatické hodnocení projektů zaměřených na SQL jazyk. Je podporované verzování nahraných verzí, přičemž systém uchovává všechny nahrané verze projektů. Studentem nahrané řešení projektu obsahuje jeho login, čas nahrání, číslo revize, název nahraného souboru. Učitel může hodnotit a komentovat pouze jemu přidělené projekty, přičemž Studentovi je umožněno prohlížet si výsledky přímo v systému a reagovat na komentáře. Ohodnocené výsledky mimo jiné evidují počet dosažených bodů, číslo hodnocené revize, čas ohodnocení a stav výsledku projektu.

Celý systém ukládá informace ohledně celého jeho chodu, jmenovitě ve které sekci událost proběhla, kdo jí vykonal, čas vykonání, název akce a případný popis vyvolané chyby.

DB Education 1.0 se skládá celkem z 22 tabulek. Návosloví těchto tabulek rozděluje databázi do 4 kategorií:

- User
- Project
- Study
- Log and system

Relační schéma této databáze lze vidět níže, viz obrázek 1.



Obrázek 1: Relační schéma databáze DB Education 1.0



## 2.4 Datová vrstva

DB Education 1.0 používá k ukládání dat relační databázový systém Microsoft SQL Server, který přináší krom ověřené funkčnosti taky užitečné výhody. Jedná se o velice spolehlivé úložiště dat s rychlým přístupem k nim. Dále Microsoft SQL Server nabízí konzistentní přístup a zabezpečení dat, prosazuje jejich pravidla integrity tak, aby byla zajištěna jejich spolehlivost a správná konzistence [9].

Pro správu a přístup k datům se využívá objektově relačního mapování pomocí frameworku Entity Framework Core, který je vyvíjen společností Microsoft. Umožňuje vývojářům v rozhraní .NET [10] pracovat s databází pomocí objektů .NET. Pro získání dat z databáze používá dotazovací jazyk LINQ [11], jenž umožňuje zapisovat dotazy silného typu. Nedílnou součástí Entity Framework Core jsou také Migrace, které umožňují během vývoje vytvořit a dynamicky měnit datový model databáze [12].

Každá databázová tabulka je reprezentována POCO třídou, též nazývané jako entity, kdy vytvořený objekt této třídy reprezentuje jeden řádek z databázové tabulky. Nedílnou součástí je DbContext, který je zodpovědný za správu spojení s databázovým systémem, kdy za pomocí dostupného API můžeme s databází objektově pracovat.

Oddělením jednotlivých vrstev informačního systému jsme schopni docílit zlepšení porozumění dané části (stane se jednodušší pro pochopení a má vymezené své hranice. Je definován jasný kontext). Taková vrstva se zároveň stává snadněji testovatelnou, rozšířitelnou a v neposlední řadě spravovatelnou. Dalším důležitým aspektem je možnost na vrstvě samostatně pracovat nezávisle na ostatních vrstvách. V tomto případě však k oddělení vrstev došlo jen z části. Datová vrstva je poměrně úzce spojena s business logikou (aplikační/business vrstva), což z pohledu udržitelnosti kódu, je poměrně problémová situace.

## 3 DB Education 2.0

### 3.1 O verzi

Od verze 1.0 se během dvou let prokázalo, že některé funkcionality systému jsou nedokonalé nebo zcela nadbytečné, například: nedostatečná autentizace uživatele, správa online testů, zápočty a jiné. Vizuální stránka se rovněž odlišuje od zavedených grafických norem definovaných v případě informačního systému DB Manager. Verze 2.0 bude tak mnohem jednodušší, přehlednější, uživatelsky přívětivější a zbavená nadbytečných funkcionalit. Verze 2.0 se navrácí ke standardnímu přístupu Request-Response v rámci využití technologie ASP.NET Core za využití návrhové vzoru MVC [13]. Větší uživatelskou interakci pak systém bude poskytovat v součinnosti s Javascriptem, případně javascriptovou knihovnou jQuery [14].

### 3.2 Funkční analýza

V této kapitole dojde k definování nových funkcí a upravené funkce jsou viditelné v tabulkách.

#### Odebrané sekce

Sekce	Důvod
3. Správa uživatelských skupin	Role se budou přiřazovat rovnou v sekci <i>Správa uživatelů</i>
10. Zápočty	Sekce je nadbytečná
11. Správa online testů	Sekce je nadbytečná
13. Nezařazené	<i>Odebírání novinek</i> je nadbytečná funkce

Tabulka 1: Odebrané sekce z DB Education 2.0

#### Upravené sekce

Sekce	Úprava
4. Správa akademických roků	Spravuje jen Administrátor
5. Správa předmětů	Spravuje jen Administrátor
6. Správa kurzů	Sjednocení pod sekci <i>Správa předmětů</i> Spravuje jen Administrátor Editace obsahu kurzu spravuje i Učitel
7. Správa statických stránek	Nový název: Správa položek menu Spravuje jen Administrátor Nová vlastnost podsložek
8. Správa projektů – specifikace	Kromě 8.5 spravuje jen Administrátor

Tabulka 2: Upravené sekce DB Education 2.0

## Změněné funkce

Sekce	Číslo funkce	Změna	Popis
2. Správa uživatelů	2.3	Úprava	- Nově také nastaví stav aktivní a přidá evidovanou roli (Administrátor, Učitel)
	2.4	Úprava	- Nově pouze upraví roli uživatele
	2.5	Odebrání	- Změnu stavu uživatele nebude řešit samostatná funkce
	2.6	Úprava	- Nově změní stav uživatele na neaktivní
4. Správa akademických roků	4.3	Úprava	- Nově umožní vytvořit i připravovaný rok
5. Správa předmětů	5.1 6.1	Úprava	- Nově se funkce sjednotí a jejich seznam bude seřazený podle pořadí
	6.5	Odebrání	- Nebude umožněno smazat předmět v akademickém roce
7. Správa položek menu	7.1	Úprava	- Nově bude seznam seřazen podle pořadí
9. Správa projektů – řešení	9.3	Úprava	- Nově změní stav projektu na základě udělených bodů, data ohodnocení a poté volá funkci 9.8
	9.4	Odebrání	- Nebude třeba měnit stav samostatnou funkcí, to se bude dít ve funkcích k tomu určených
	9.5	Úprava	- Nově nebude možno vkládat poznámky
12. Logování	12.3	Odebrání	- Nadbytečná funkce
	12.4	Úprava	- Logy se budou ukládat pouze do databáze

Tabulka 3: Změněné funkce DB Education 2.0

Se systémem stále pracují tři typy uživatelů: Administrátor, Učitel a Student. Systém má taktéž svou vlastní roli.

#### **4. Správa akademických roků**

Role: Administrátor

- 4.6. Aktivace akademického roku – změni stav vybraného roku na aktivní a zároveň změni stav předchozího aktivního roku na neaktivní a s tím i všechny specifikace projektů přiřazené k tomuto roku.

#### **5. Správa předmětů**

Role: Administrátor

- 5.7. Přehození pořadí předmětů v akademickém roce – prohodí pořadí 2 předmětů v akademickém roce.
- 5.8. Kopie obsahu z minulého akademického roku – vrátí obsah předmětu z předchozího akademického roku.

#### **7. Správa položek menu**

Role: Administrátor

- 7.6. Přehození pořadí položek menu – přehodí pořadí 2 položek menu.
- 7.7. Editace obsahu položky menu – upraví pouze obsah položky menu.

#### **8. Správa projektů – specifikace**

Role: Administrátor

- 8.5. Přidání deadlinu ke specifikaci projektu – k vybrané specifikaci projektu vytvoří deadline.
- 8.6. Smazání deadlinu ze specifikace projektu – z vybrané specifikace projektu odstraní deadline.
- 8.7. Seznam deadlinů – vrátí seznam deadlinů vybrané specifikace projektu.

#### **9. Správa projektů – řešení**

Role: Administrátor, Učitel

- 9.15. Vrácení projektu k dopracování – změni stav projektu k dopracování a poté volá funkci 9.8.

#### **14. Systémové nastavení**

Role: Administrátor

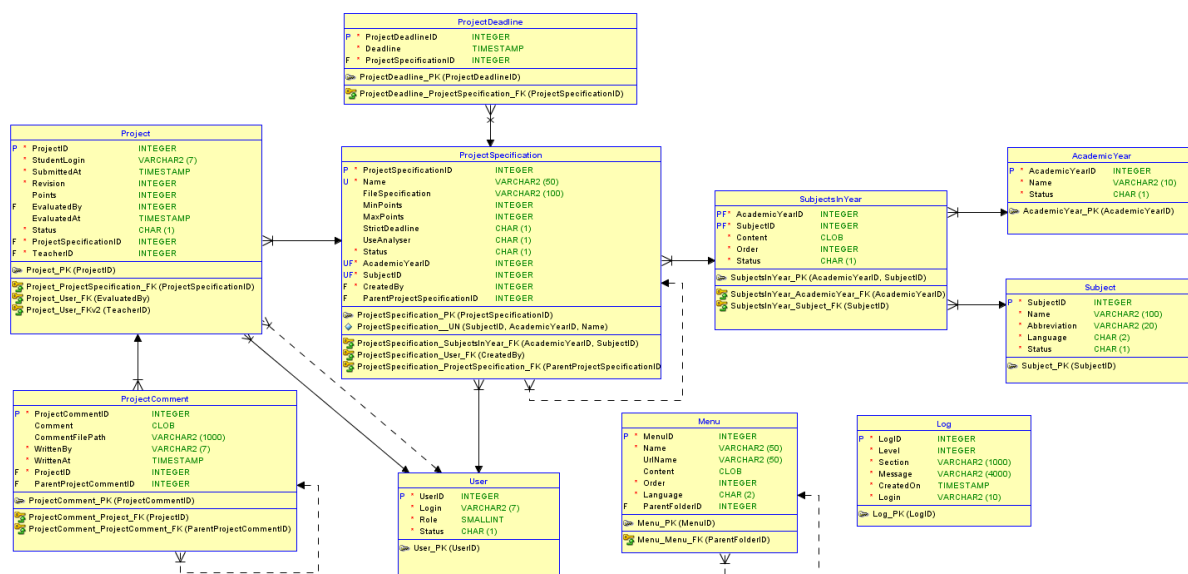
- 14.1. Úprava systémového nastavení – uloží změny systémového nastavení, například: adresa SMTP a LDAP serveru, cesta adresáře pro ukládání projektů nebo obsahu položek menu.

### 3.3 Datová analýza

Na základě provedených změn dochází v databázi k následujícím úpravám:

1. Zrušení přerozdělení čtyřech skupin a odstranění prefixu z názvu tabulek.
2. Sjedení tabulek původní skupiny *User* do jedné stejnojmenné tabulky, již bude uchovávat pouze atributy login, role a stav aktuálnosti. Vedený uživatel systému má pouze jednu roli a nadále nefiguruje ve skupinách.
3. Přejmenování tabulky *st\_static\_page* na *Menu*, odebrání atributu status, přidání nových atributů pořadí a cizího klíče odkazující se na primární klíč samotné tabulky. Důvodem přejmenování je dosažení lepší přehlednosti, to platí i u dalších přejmenování. Statická stránka nově může dynamicky měnit svoje pořadí a je schopné sdružovat pod sebou další přiřazené statické stránky.
4. Důsledkem odstraněných sekcí ze skupiny *Study* dojde k odstranění těchto tabulek:
  - *st\_test\_result*
  - *st\_test*
  - *st\_news*
  - *st\_registration\_test*
  - *st\_registration\_test\_user*
  - *st\_subject\_academics\_year\_newsletter*
5. Přejmenování tabulky *st\_subject\_academic\_year* na *SubjectsInYear*, odstranění primárního klíče a vytvoření složeného primárního klíče z akademického roku a předmětu, odebrání atributů *study\_form* a *semestr\_type*, přidání nového atributu pořadí. Předměty v akademickém roce se v systému nerozdělovaly ani podle typu studia ani na semestry, avšak vlastnost měnit pořadí položek v menu předchozí verze systému strádala.
6. Přejmenování tabulky *st\_academic\_year* na *AcademicYear*.
7. Přejmenování tabulky *st\_subject* na *Subject*.
8. Odstranění tabulky *log\_auth\_key*.
9. Přejmenování tabulky *log\_system* na *Log*, odebrání atributu status.
10. Sjedení tabulek *pr\_task\_result* a *pr\_task\_upload* do tabulky *Project*, odstranění atributů *note*, *name\_file*, *study\_form\_upload* a *is\_final*, přidání atributu cizího klíče odkazující se na hodnotícího uživatele. Sjedením se dosáhne ke zjednodušení a ke snížení množství uložených záznamů v databázi. Záznam nahraného projektu tak může být i rovnou opraven a není nutné vytvářet nový záznam v jiné tabulce, který se nepřímo spojí se záznamem nahraného projektu. Název nahraného souboru se dynamicky vytvoří z loginu studenta a pořadí nahrané revize.
11. Přejmenování tabulky *pr\_task* na *ProjectSpecification*, odstranění atributu *study\_form*. Forma studia se nevyužívala.
12. Přejmenování tabulky *pr\_task\_deadline* na *ProjectDeadline*.
13. Přejmenování tabulky *pr\_task\_upload\_comment* na *ProjectComment*, odstranění atributu status, přidání atributu cesty nahraného souboru. Komentář se již nerozděluje jako poznámka.

Takto upravená databáze nyní obsahuje pouze 10 tabulek, což je oproti DB Education 1.0 razantní změna. Relační schéma upravené databáze DB Education 2.0 lze vidět níže, viz obrázek 2.



Obrázek 2: Relační schéma databáze DB Education 2.0

### 3.4 Datová vrstva

DB Education 2.0 nadále pracuje s databázovým systémem Microsoft SQL Server. Důvodem zachování databázového systému je ověřená funkčnost a výkon tohoto systému.

Oproti původní verzi je zde aplikována vícevrstvá architektura. Datová vrstva je v projektu nyní oddělená od aplikační části.

Optimalizovaný systém i nadále využívá pro správu a přístup k datům ORM Entity Framework Core. Pro práci s databází jsou mimo jiné použité i takzvané Query objekty. Jedná se o zapouzdřené třídy, které jsou vymezené vůči vlastnímu kontextu. Následně objekty vytvořené z těchto tříd reprezentují právě jednu databázovou operaci. Takový objekt je separován od ostatních dotazů a má jednoznačně určené pole své působnosti. Veškeré své závislosti, nutné k úspěšnému provedení operace, získává na začátku svého vytvoření. Objekt obsahuje jen data, které potřebuje ke svému chodu, o ostatních operacích nemá tušení. Zmíněné vlastnosti přináší řadu výhod. Objekty nemají komplikovanou logiku, jsou přehlednější a snadněji testovatelné. Dají se lehce rozšířit o nové chování. Spravovatelnost kódu je přímější a jednodušší. Každý Query objekt pak dědí z třídy *BaseQuery*, jenž implementuje synchronní a asynchronní metody pro spuštění těchto dotazů. Pro možnost vytvoření parametrizovaného dotazu je k dispozici třída *ParametrizedQuery*, která umožňuje použití vstupních parametrů.

LINQ v mnoha případech usnadňuje tvorbu databázových dotazů, ovšem někdy je jeho použití neefektivní. V takových momentech je vhodné využít vlastnosti LINQ, a to spuštění SQL raw dotazů [15]. Této vlastnosti se konkrétně využívá u rekurzivního CTE dotazu, kdy je potřeba získat hierarchii položek menu.

## 4 DB Manager 1.0

### 4.1 O verzi

Informační systém DB Manager 1.0 byl v roce 2013 vytvořen Ing. Martinem Zwierzynou jako bakalářská práce [16]. V roce 2015 tento systém rozšířil a upravil v diplomové práci [17]. Verze systému byla navržena za pomoci ASP.NET [18], který vytváří webové stránky díky takzvaným webovým formulářům. Verze systému dokáže spolupracovat s databázovými servery jako je Microsoft SQL Server, Oracle, MySQL, PostgreSQL a IBM DB2 [19]. Databázové účty se pravidelně synchronizují s účty vedených přímo na databázových serverech a naopak.

### 4.2 Funkční analýza

Se systémem budou pracovat tři typy uživatelů: Systémový administrátor, Databázový administrátor a VŠB osoba. Systém bude mít taktéž svou vlastní roli.

- *Systémový administrátor*  
Stará se o chod celého systému. V jeho kompetenci je správa uživatelských rolí, správa a konfigurace databázových serverů, vyřizování žádostí uživatelů a taktéž může spravovat databázové účty na všech databázových serverech.
- *Databázový administrátor*  
Spravuje databázové účty uživatelů na databázových serverech, které mu přiřadil Administrátor. Mezi jeho pravomoci patří také vyřizování žádostí uživatelů.
- *VŠB osoba*  
Má možnost prohlížet si seznam veřejných serverů, posílat na ně žádosti pro vytvoření účtu nebo si může prohlížet vlastní účty a posílat žádosti o změnu jejich parametrů anebo resetovat jejich heslo.
- *Systém*  
Specializovaná role, která má v kompetenci pravidelné spouštění plánovaných úloh, posílání emailů a logování celého systému.

### 1. Autentizace

- 1.1. Přihlášení do systému – Uživatelé jsou přihlášení do systému za použití univerzitních údajů ověřených prostřednictvím LDAP protokolu.
- 1.2. Odhlášení ze systému – Uživatelé jsou odhlášení ze systému.

### 2. Správa uživatelských rolí

Role: Systémový administrátor

- 2.1. Přidání role – přidá uživateli roli systémového nebo databázového administrátora a volá funkci 11.1. V případě databázového administrátora přidá i přiřazený databázový server.
- 2.2. Odebrání role – odebere uživateli roli a volá funkci 11.1.
- 2.3. Detail uživatele – vrací detailní popis vybraného uživatele
- 2.4. Seznam uživatelů – vrací seznam všech aktivních uživatelů systému.

### 3. Správa databázových účtů

Role: Systémový administrátor, Databázový administrátor

- 3.1. Vytvoření databázového účtu – vytvoří na vybraném databázovém serveru databázi a databázový účet k této databázi s možností ji spravovat. Poté volá funkci 11.1, kde se předají údaje k připojení.
- 3.2. Hromadné vytvoření databázových účtů – prochází seznam loginů, kterým vytváří databázové účty pomocí funkce 3.1.
- 3.3. Filtrace databázových účtů – vrací seznam databázových účtů filtrovaných podle vybraných atributů.
- 3.4. Smazání databázového účtu – smaže databázi s přiřazeným databázovým účtem a volá funkci 11.1.
- 3.5. Změna parametrů databázového účtu – upraví kvótu nebo expiraci vybraného databázového účtu.
- 3.6. Záloha databázových účtů – provede zálohu vybraného databázového účtu.

Role: Systémový administrátor, Databázový administrátor, VŠB osoba

- 3.7. Detail databázového účtu – vrací detailní popis vybraného databázového účtu.
- 3.8. Reset hesla – změní heslo vybraného databázového účtu.

### 4. Správa databázových serverů

Role: Systémový administrátor

- 4.1. Přidání databázového serveru – vytvoří nový záznam databázového serveru.
- 4.2. Úprava databázového serveru – upraví parametry záznamu vybraného databázového serveru.
- 4.3. Odebrání databázového serveru – odebere vybraný záznam databázového serveru, který nemá přiřazené žádné databázové účty v tomto informačním systému.
- 4.4. Vytvoření procedur na databázovém serveru – spustí na vybraném databázovém serveru základní procedury potřebné ke správě databázových účtů.
- 4.5. Test existence procedur – zjistí existenci základních procedur na vybraném databázovém serveru.
- 4.6. Test připojení serveru – pokusí se připojit na vybraný databázový server a vrátí odpověď, jestli se to povedlo, nebo ne.
- 4.7. Seznam databázových serverů – vrací seznam databázových serverů.
- 4.8. Detail databázového serveru – vrací detail vybraného databázového serveru.



## 5. Správa žádostí

Role: Systémový administrátor, Databázový administrátor

- 5.1. Schválení žádosti – upraví nebo vytvoří databázový účet na základě požadavku žádosti a volá funkci 11.1.
- 5.2. Zamítnutí žádosti – zamítne žádost s důvodem zamítnutí a volá funkci 11.1.
- 5.3. Úprava žádosti – upraví žádané parametry vybrané žádosti.
- 5.4. Seznam žádostí – vrací seznam zpracovaných nebo nezpracovaných žádostí.
- 5.5. Detail žádosti – vrací detail vybrané žádosti.

Role: VŠB osoba

- 5.6. Podání žádosti o vytvoření – vytvoří novou žádost o vytvoření databázového účtu a volá funkci 11.1, která informuje jednoho z administrátorů databázového serveru nebo systémového administrátora.
- 5.7. Podání žádosti o změně parametrů – vytvoří novou žádost o změně parametrů databázového účtu a volá funkci 11.1, která informuje jednoho z administrátorů databázového serveru nebo systémového administrátora.

## 6. Logování

Role: Systémový administrátor, Databázový administrátor

- 6.1. Výpis logů – zobrazí seznam všech logů.
- 6.2. Detail logu – zobrazí detailní informace vybraného logu.

Role: Systém

- 6.3. Přidání logu – vytvoří nový log záznam systému a zapisuje jej do XML souboru.

## 7. Správa systémových služeb

Role: Systém

- 7.1. Synchronizace – prochází všechny databázové servery a jejich neignorované databázové účty, které porovnává s databázovými účty vedenými v informačním systému. Jestliže se databázový účet nevyskytuje na databázovém serveru nebo v informačním systému, vloží se záznam o takovém účtu do tabulky synchronizace.
- 7.2. Test připojení – prochází všechny databázové servery a pokouší se na ně připojit do té doby, než se naváže spojení. Počet opakování a dobu prodlevy určují předem definované hodnoty z konfigurace. Pokud se u některého z databázových serverů nepovede připojit, volá funkci 11.1, která informuje vlastníka databázového serveru.
- 7.3. Kontrola platnosti účtů – V půli každého měsíce prochází jednotlivé databázové účty a kontroluje jejich expiraci. Vlastníkům, jejichž platnost účtu expiruje na konci daného měsíce, jsou pak posílány informační emaily (funkce 11.1). Na konci každého měsíce jsou pak expirované účty smazány a jejich vlastník je o tom informován (funkce 11.1).

- 7.4. Zálohování – prochází všechny databázové účty s příznakem zálohovat. Tyto účty se zálohují na daném databázovém serveru a jejich vlastník je informován (funkce 11.1) o provedené záloze.

## **8. Statistiky**

Role: Systémový administrátor

- 8.1. Využití databázových serverů – vrací seznam databázových serverů a jejich počet vytvořených databázových účtů v informačním systému.
- 8.2. Využití systému – vrací předem definovaný seznam provedených operací systému a jejich počet, například: počet vytvořených databázových účtů, počet schválených žádostí.

## **9. Konfigurace**

Role: Systémový administrátor

- 9.1. Uložení konfigurace – uloží základní nastavení systému například: délka generovaného hesla, adresa SMTP a LDAP serveru.
- 9.2. Nastavení plánovače – uloží data s časy plánovaných služeb systému.

## **10. Synchronizace**

Role: Systémový administrátor, Databázový administrátor

- 10.1. Vložení ignorovaného účtu – přidá název databázového účtu daného databázového serveru do ignorované výjimky procesu synchronizace.
- 10.2. Odebrání ignorovaného účtu – odebere název databázového účtu daného databázového serveru z ignorované výjimky procesu synchronizace.
- 10.3. Spuštění synchronizace – funguje stejně jako funkce 7.1 s tím rozdílem, že se nespouští automaticky.
- 10.4. Smazání nesynchronizovaného účtu – smaže databázový účet z informačního systému anebo z databázového serveru, podle toho, kde se daný databázový účet nachází.
- 10.5. Vložení nesynchronizovaného účtu – vytvoří databázový účet v informačním systému anebo na databázovém serveru, podle toho, kde se daný databázový účet nenachází.
- 10.6. Seznam nesynchronizovaných účtů – vrátí seznam nesynchronizovaných účtů.

## **11. Ostatní**

Role: Systém

- 11.1. Odeslání informačních zpráv – posílá informační zprávu uživateli prostřednictvím SMTP klienta.

### 4.3 Datová analýza

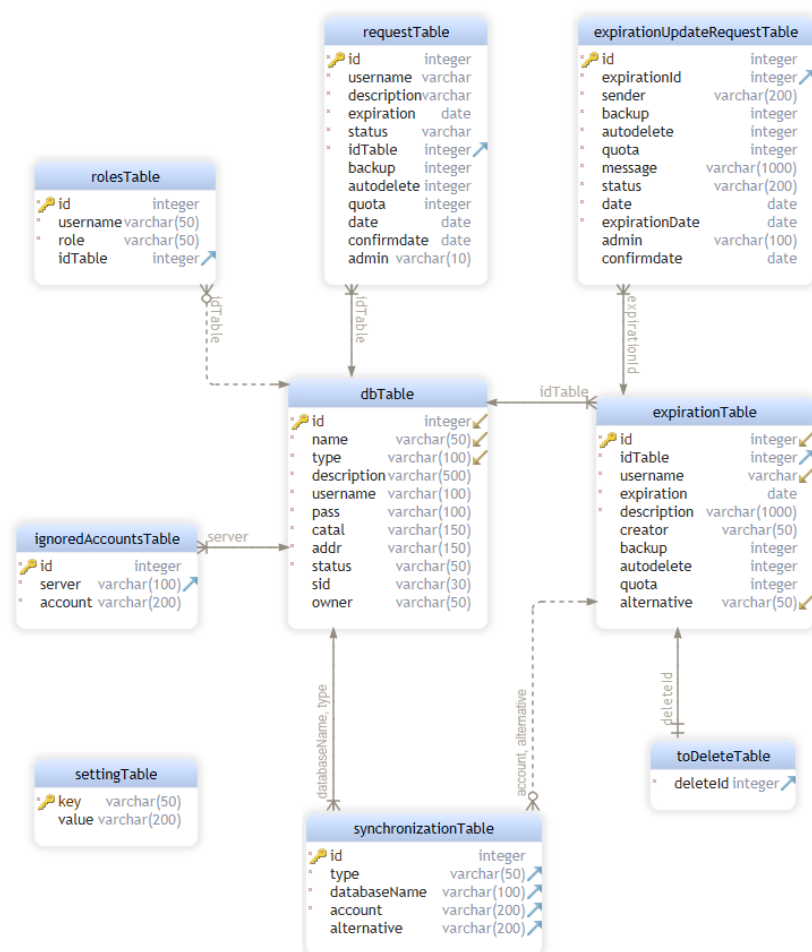
Základem celého systému je správa databázových serverů, kde se uchovává název serveru, popis, typ serveru (např. Oracle, Microsoft SQL Server, ...), parametry k připojení a kteří správci se o server starají včetně jednoho hlavního správce, jenž bývá informován emailem o hlavních událostech serveru.

Na daných databázových serverech jsou spravovány databázové účty, u kterých se uchovává login vlastníka a uživatele, který tento účet vytvořil, popis, kvótu, dobu platnosti a taktéž informace, jestli se má účet zálohovat a automaticky smazat po skončení platnosti. V případě, že by uživatel chtěl více účtů na jednom serveru, tak se bude uchovávat i alternativní název účtu.

Systém také umožňuje spravovat žádosti o vytvoření nového účtu nebo žádosti o změně stávajícího účtu. U žádosti o vytvoření nového účtu je žádoucí uchovávat login uživatele, popis, parametry vytvářeného účtu (doba platnosti, kvóta, ...), čas vytvoření, stav žádosti a databázový server, kde se má vytvořit. U žádosti o změně stávajícího účtu je také potřebné uchovávat login uživatele, popis, změněné parametry účtu, čas vytvoření a stav žádosti. Po zpracování žádostí uchovávají mimo již zmíněných hodnot také čas zpracování a login uživatele, který žádost zpracoval a případně jeho důvod zamítnutí.

Systém definuje klíče a jejich hodnoty, které vymezují čas spuštění plánovaných služeb (synchronizace, zálohování, ...) nebo konfigurují délku generovaného hesla databázových účtů. V rámci synchronizace se uchovávají názvy účtů, jež nebyly zaznamenány v systému, nebo na serveru. Databázové účty, které se nemají synchronizovat (např. systémové účty) je možné zadefinovat a při další synchronizaci tak budou ignorovány.

DB Manager 1.0 se skládá celkem z 9 tabulek. Relační schéma této databáze lze vidět níže, viz obrázek 3.



Obrázek 3: Relační schéma databáze DB Manager 1.0

#### 4.4 Datová vrstva

DB Manager 1.0 používá k ukládání dat databázový systém SQLite [20]. SQLite je malá knihovna nástrojů, které implementují relační databázový systém. Každá databáze se ukládá jako jeden obyčejný soubor na disk a je díky tomu přenositelná. Z tohoto důvodu se hojně využívá [21]. Pro práci s logy systém pracuje s XML [22] souborem. XML je značkovací jazyk, jenž umožňuje strukturovaně ukládat a číst data z takového souboru.

Pro správu a přístup k datům není použitý žádný ORM framework, nýbrž byla použita vlastní implementace. Tato implementace se nachází v samostatné datové vrstvě. Pro správu a přístup k datům relační databáze se využívá sada tříd ADO.NET [23], konkrétně pak třídy zaměřené na práci s databází SQLite. Veškerá tato logika je implementována v samostatné třídě. Objekt této třídy obsahuje metody pro připojení k databázi a metody pro vykonávání samotných příkazů posílaných na databázi. Metody, které vrací data, získaných z dotazů, se mapují do POCO tříd. Tyto třídy se nachází v samostatné podvrstvě *ListEntities*. Objekt těchto tříd reprezentují jeden řádek databáze.

V datové vrstvě se také nacházejí třídy pro práci s jednotlivými databázovými servery, jimiž jsou spravovány v informačním systému. Tyto třídy implementují rozhraní *IConnections*, které obsahuje stanovené chování určené pro připojení k databázovému serveru a k vykonání příkazu na serveru. Každý objekt třídy databázového systému, za pomoci sady tříd ADO.NET, implementuje vlastní zpracování chování metod. Metody jsou popsány níže, viz tabulka 4.

Název metody	Popis
CreateLogin	Spustí proceduru pro vytvoření účtu, databázi, případně i kvótu, jestli je podporována
DeleteLogin	Spustí proceduru pro smazání účtu a databázi
SetQuota	Spustí proceduru pro nastavení kvóty
GetQuota	Vrátí hodnotu nastavené kvóty
GrantConnect	Přiřadí práva k připojení k účtu
DenyConnect	Odebere práva k připojení k účtu
GetLogins	Vrátí seznam uživatelských účtů
GetLoginsPermissions	Vrátí seznam uživatelských účtů, které mají oprávnění se připojit
CheckLogin	Ověří existenci účtu
CheckPermission	Ověří oprávnění účtu
CheckLock	Ověří uzamknutí účtu
UnlockAccount	Odemkne účet
PasswordReset	Změní heslo účtu
CreateProcedure	Vytvoří základní procedury na serveru
CheckProcedures	Ověří existenci základních procedur na serveru

*Tabulka 4: Metody rozhraní IConnections*

## 5 DB Manager 2.0

### 5.1 O verzi

Od nasazení verze 1.0 uplynulo pět let a za tu dobu se změnilы trendy designu, technologií a potřeby učitelů databázových předmětů. Prokázaly se taktéž nedostatky této verze, například: absence odemknutí přihlášení k databázovým účtům na serverech, chybějící transakce nebo značné zpomalování ve vyhledávání logů. Verze 2.0 tyto nedostatky opraví a celý systém zefektivní, zjednoduší, zpřehlední, přinese nové funkcionality a zavede responzivní design, podle aktuálních trendů. Verze 2.0 již nadále nebude využívat původní koncepci webových formulářů, ale místo ní použije koncepci MVC, která za použití technologie ASP.NET Core MVC nabízí rozdělení webu na logiku a výstup, jež spolu propojuje pomocí kontroléru.

### 5.2 Funkční analýza

V této kapitole dojde k definování nových funkcí a upravené funkce jsou viditelné v tabulkách.

#### Upravené sekce

Sekce	Úprava
6. Logování	Spravuje jen Systémový administrátor
9. Konfigurace	Nový název: Systémové nastavení

*Tabulka 5: Upravené sekce DB Manager 2.0*

## Změněné funkce

Sekce	Číslo funkce	Změna	Popis
2. Správa uživatelských rolí	2.1	Úprava	- Nově umožní přidat uživateli více než jen jeden spravovaný server
	2.2	Úprava	- Nově odebere i spravované servery
	2.3	Úprava	- Nově k detailu vrací i seznam spravovaných databázových serverů
3. Správa databázových účtů	3.6	Odebrání	- Zálohování se bude spouštět pouze přes plánovanou službu
5. Správa žádostí	5.3	Úprava	- Nově upravenou žádost rovnou schválí - Přejmenování na <i>Schválení žádosti se změnou</i>
6. Logování	6.3	Úprava	- Nově se log záznamy zapisují do databáze
7. Správa systémových služeb	7.1	Úprava	- Nově prochází pouze databázové účty vedené v informačním systému a hledá jejich výskyt na daném databázovém serveru. - Nenalezené účty získají příznak nesynchronizovaný.
9. Systémové nastavení	9.1 9.2	Úprava	- Nově jsou funkce sjednocené do jedné a systémové nastavení se ukládá do JSON [24] souboru
10. Synchronizace	10.1	Odebrání	- Ignorované účty nebudou nadále figurovat v systému
	10.2	Odebrání	- Ignorované účty nebudou nadále figurovat v systému
	10.3	Odebrání	- Synchronizace se bude spouštět pouze přes plánovanou službu.
	10.4	Úprava	- Nově smaže záznam nesynchronizovaného databázového účtu z informačního systému.
	10.5	Úprava	- Nově vytvoří nesynchronizovaný účet na databázovém serveru a volá funkci 11.1, která informuje vlastníka účtu.

Tabulka 6: Změněné funkce DB Manager 2.0

Se systémem stále pracují tři typy uživatelů: Systémový administrátor, Databázový administrátor a VŠB osoba. Systém bude má taktéž svou vlastní roli.

## **2. Správa uživatelských rolí**

Role: Systémový administrátor

- 2.5. Odebrání databázových serverů správce databáze – odebere vybrané databázové servery vybraného správce databáze a informuje ho o této skutečnosti (funkce 11.1).
- 2.6. Seznam databázových serverů k přiřazení – vrátí seznam databázových serverů, které správce databáze může spravovat.

## **3. Správa databázových účtů**

Role: Systémový administrátor, Databázový administrátor, VŠB osoba

- 3.9. Kontrola uzamknutí databázového účtu – zkontroluje, jestli je účet uzamknutý na databázovém serveru.
- 3.10. Odemknutí databázového účtu – zpřístupní přihlášení účtu na databázovém serveru.

Role: Systémový administrátor, Databázový administrátor

- 3.11. Hromadné prodloužení platnosti databázových účtů – hromadně prodlouží platnost všem vybraným databázovým účtům.
- 3.12. Hromadné smazání databázových účtů – hromadně smaže vybrané databázové účty.

## **7. Správa systémových služeb**

Role: Systémový administrátor, Databázový administrátor

- 7.5. Seznam plánovaných služeb – vrátí seznam plánovaných služeb systému s informací jejich posledního spuštění.

V této verzi byla přidána následující sekce:

## **12. Notifikace**

Role: Systémový administrátor, Databázový administrátor

- 12.1. Kontrola notifikací – informuje, jestli je v systému:
  - a) Nesynchronizovaný účet
  - b) Expirovaný účet
  - c) Nefunkční databázový server
  - d) Nevýřízená žádost

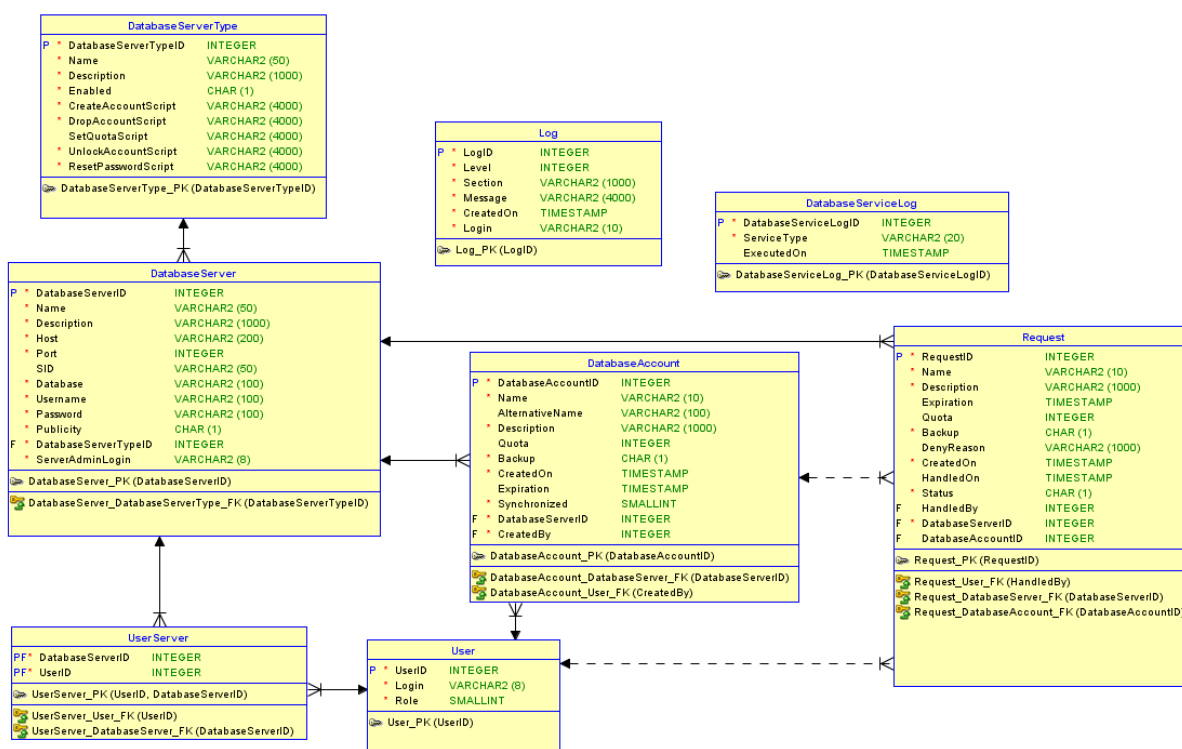


### 5.3 Datová analýza

Na základě provedených změn dochází v databázi k následujícím úpravám:

14. Přejmenování tabulky *rolesTable* na *User*. Důvodem přejmenování je dosáhnout lepší přehlednosti, to platí i níže.
15. Přidání nové tabulky *DatabaseServerType*, jíž definuje typ databázového serveru. Každý typ databázového serveru uchovává jeho název, popis, stav použitelnosti a uložené procedury pro vytváření, smazání, odblokování účtu nebo také pro nastavení kvóty či resetování hesla účtu.
16. Přejmenování tabulky *dbTable* na *DatabaseServer*, rozdělení atributu adresy serveru na 2 samostatné atributy *host* a *port*, odebrání atributu *type*, přidání cizího klíče tabulky *DatabaseServerType*.
17. Přidání nové vazební tabulky *UserServer* obsahující cizí klíče tabulky *User* a *DatabaseServer*.
18. Přidání nové tabulky *Log* pro ukládání informací ohledně chodu celého systému. Každá informace uchovává jmenovitě ve které sekci událost proběhla, kdo jí vykonal, čas vykonání, název operace a případný popis vyvolané chyby.
19. Přidání nové tabulky *DatabaseServiceLog* uchovávající čas úspěšně vykonaných plánovaných služeb systému.
20. Odstranění tabulky *settingTable*. Nastavitelné hodnoty se nově budou ukládat a načítat ze souboru JSON.
21. Sjednocení tabulek *requestTable* a *expirationUpdateRequestTable* do tabulky *Request*, odstranění atributu *autodelete*. Důvodem je zjednodušení a fakt, že oba typy žádostí mají téměř stejné atributy.
22. Přejmenování tabulky *expirationTable* na *DatabaseAccount*, odebrání atributu *autodelete*, přidání nového atributu stavu synchronizace. Expirované účty se automaticky smažou při plánované službě *kontroly platnosti účtů* a fakt, jestli je účet synchronizovaný nebo nesynchronizovaný, se určí nově přidaným atributem.
23. Odstranění následujících tabulek:
  - *synchronizacionTable*
  - *ignoredAccountsTable*
  - *toDeleteTable*
24. Upravení délek velikostí datových typů.

Takto upravená databáze nyní obsahuje 8 tabulek se srozumitelnějšími názvy atributů. Relační schéma upravené databáze DB Manager 2.0 lze vidět níže, viz obrázek 4.



Obrázek 4: Relační schéma databáze DB Manager 2.0

## 5.4 Datová vrstva

DB Manager 2.0 používá databázi relačního databázového serveru Microsoft SQL Server.

Na rozdíl od předchozí verze systém využívá pro správu a přístup k datům ORM Entity Framework Core, který je popsán v kapitole 2.4. Jednotlivé databázové operace jsou rozdělené do takzvaných *Controllerů*. Každý *Controller* má svou vlastní logiku a spravuje sadu extension metod, které jsou spojené s konkrétní databázovou tabulkou.

Pro migraci dat, je vytvořena třída za pomoci Entity Framework Core, ve které je vytvořený seed, který se použije. Mechanismus seedování zahrnuje i vložení dat do databázové tabulky *DatabaseServerType*. Tyto data reprezentují využitě databázové typy čtyř databázových systémů a obsahují základní skripty pro vytvoření procedur.

Stejně jak tomu bylo i v předchozí verzi 1.0, tak se i zde nacházejí třídy pro obsluhu samotných databázových serverů. Jsou to třídy *MsSqlManager*, *MySqlManager*, *OracleManager* a *PosgreSQLManager*. Tyto třídy dědí z *BaseManageru*. *BaseManager* poskytuje jednotné rozhraní pro přístup a práci s databázovými servery. Každý databázový manager si tak zapouzdřuje svou vlastní logiku pro komunikaci se svým databázovým serverem. V programu se pak manager dynamicky alokuje za běhu, a to podle vlastností konkrétního databázového serveru. Ze vstupních parametrů databázového serveru dochází k sestavování připojovacího řetězce, taktéž za běhu programu. Připojovací řetězec umožňuje navázat komunikaci mezi managerem a databázovým serverem. Jednotliví manageři reprezentují svou vlastní logiku pro komunikaci s databázovými servery pomocí

svých metod. Tyto metody vycházejí z předchozí verze 1.0 (popsané v tabulce 4), avšak jsou optimalizované a přehled jejich změn je popsán níže, viz tabulka 7.

<b>Původní název</b>	<b>Nový název</b>	<b>Popsaná změna</b>
CreateLogin	CreateDatabaseAccount	- Přejmenování metody
DeleteLogin	DropDatabaseAccount	- Přejmenování metody
SetQuota	SetQuota	- Dostupné jen pro <i>OracleManager</i> a <i>MsSqlManager</i>
GetQuota	-	- Metoda odebrána
GrantConnect	-	- Metoda odebrána
DenyConnect	-	- Metoda odebrána
GetLogins	-	- Metoda odebrána
GetLoginsPermissions	DatabaseAccountExists	- Přejmenování metody
CheckLogin	UserExists	- Přejmenování metody
CheckPermission	UserHasRights	- Přejmenování metody
CheckLock	IsDatabaseAccountLocked	- Přejmenování metody
UnlockAccount	UnlockDatabaseAccount	- Spustí proceduru pro odemknutí databázového účtu - Přejmenování metody
PasswordReset	ResetPassword	- Využití transakcí - Spustí proceduru pro reset hesla, která v případě uzamknutého účtu odemkne účet - Přejmenování metody
CreateProcedure	CreateProcedures	- Využití transakcí - Přejmenování metody
CheckProcedures	ProceduresExist	- Přejmenování metody

Tabulka 7: Optimalizované metody DB Managera 2.0

## 6 Konvertor dat

Jednoduchý program, který převádí data starého modelu do nového modelu.

Popis fungování metody pro převod dat:

1. Metoda naváže spojení s databází obsahující data starého modelu a poté naváže spojení s databází nového modelu.
2. Načtou se všechny potřebná data a převede jednotlivých seznamů objektů.
3. Vkládání do tabulek proběhne v určitém logické pořadí. Nejdříve je potřeba převést data do tabulek, které nemají žádné závislosti na ostatních tabulkách. Až potom je možné vložit záznam do tabulky, která je závislá vůči další tabulce.
4. Pro mapování primárních klíčů se využívá datová struktura Dictionary.

Ukázka kódu vyobrazuje proces převodu dat tabulek Subject a SubjectsInYear. Na řádce číslo 1 získáme všechny staré záznamy z tabulky Subject. Na řádce 4-17 procházíme seznam získaných objektů. Na řádce číslo 6 uvnitř smyčky dochází k převodu ze starého objektu na nový objekt, podporující schéma nové databáze. Na řádce 15 dojde k zavolání metody *SaveChanges()* a uložení nově převedených objektů databáze. Na řádce 20 probíhá podobný proces, kde je mimo jiné potřeba přidat i nový atribut pořadí. Na řádce 25 je podmínka, která zajistí vynulování pořadí v případě, kdy se předchozí akademický rok liší s následujícím. Na řádce 44 dojde k vložení nově převedených objektů databáze.

```
1. var st_subjects = new st_subject().GetAll(oldDbContext.st_subjects).ToList();
2. //Subject
3. var SubjectIDs_Old_New = new Dictionary<int, int>();
4. foreach (var i in st_subjects)
5. {
6.     var subject = new SubjectEntity
7.     {
8.         Status = (SubjectStatus) Enum.Parse(typeof(SubjectStatus),
9.         i.status.ToString()),
10.        Abbreviation = i.abbreviation,
11.        Language = i.lang.ToString(),
12.        Name = i.name
13.    };
14.    newDbContext.Subjects.Add(subject);
15.    newDbContext.SaveChanges();
16.    SubjectIDs_Old_New.Add(i.id_st_subject, subject.SubjectID);
17. }
18.
19. //SubjectsInYear
20. var subjectsInYear = new List<SubjectsInYearEntity>();
21. var order = 0;
22. var academicYearId = 0;
23. foreach (var i in st_subject_academic_years)
24. {
25.     if (academicYearId != i.st_academic_year_id)
26.     {
27.         academicYearId = i.st_academic_year_id;
28.         order = 0;
29.     }
30.
31.     subjectsInYear.Add(new SubjectsInYearEntity
32.     {
33.         Status = (SubjectsInYearStatus) Enum.Parse(typeof(SubjectsInYearStatus),
34.         i.status.ToString()),
35.         AcademicYearID =
```

```

36.         AcademicYearIDs_Old_New.GetValueOrDefault(i.st_academic_year_id),
37.         SubjectID = SubjectIDs_Old_New.GetValueOrDefault(i.st_subject_id),
38.         Content = i.content,
39.         Order = ++order
40.     });
41. }
42.
43. newDbContext.AddRange(subjectsInYear);
44. newDbContext.SaveChanges();

```

*Výpis 1: Ukázka převodu dat DB Education*

Princip převodu dat je pro oba systémy DB Education a DB Manager stejný.

## 7 Aplikační a prezentační vrstva

V mnohých směrech není možné sloučit aplikační a prezentační vrstvu starých systémů s novými návrhy. V systémech 2.0 tak dojde ke kompletní reimplementaci jednotlivých vrstev, ale to už je mimo rozsah této bakalářské práce.

## 8 Závěr

V této bakalářské práci jsme se zabývali optimalizováním webových informačních systémů DB Education a DB Manager. Na počátku jsme provedli důkladnou analýzu funkcionalit původních systémů. Na základě provedené analýzy jsme detekovali nedostatky. Poté jsme našli možné cesty, jak tyto nedostatky vylepšit. Vytvořili jsme aktualizovanou analýzu reflektující změny, které jsme byli schopni detekovat díky původní analýze. Vznikla také potřeba přesunout stará data na nové a k tomuto účelu jsme vytvořili konvertor dat. Pak jsme začali implementovat datovou vrstvu za využití ORM Entity Framework Core pro nové systémy, které vylepšují chyby starých systémů. Implementaci jsme poté řádně otestovali. Požadovaných výsledků se nám podařilo dosáhnout, avšak z důvodů nesoučinnosti optimalizované datové vrstvy s aplikační/prezentační vrstvou nebylo možné tyto optimalizované verze nasadit do provozu.

Práce byla velmi přínosná a naučná. Rozšířili jsme si obzor možností použití různých databázových systémů. Zdokonalili jsme si možnosti, jak lépe definovat model databáze. Naučili jsme se novým technologiím Entity Framework Core, LINQ a osvojili si práci ve vývojovém prostředí Visual Studio [25].

Pro plné nasazení optimalizovaných informačních systémů potřebujeme reimplementovat aplikační a prezentační vrstvy a po této reimplementaci budou systémy nasazené. Po nasazení do provozu bude možné tyto informační systémy podle potřeby rozšiřovat jak o nové funkcionality, tak i o nově zavedené databázové systémy pro informační systém DB Manager. Bereme fakt, že reálný provoz informačních systémů odhalí, i přes veškerou snahu, některé další nedostatky. Je proto potřeba i nadále tyto systémy aktualizovat a opravovat.

## Literatura

- [1] *Oracle Database* [online]. [cit. 2021-04-13]. Dostupné z: <https://www.oracle.com/cz/database/>
- [2] *Microsoft SQL Server* [online]. [cit. 2021-04-13]. Dostupné z: <https://www.mssql.cz/>
- [3] *MySQL | The world's most popular open source database* [online]. [cit. 2021-04-13]. Dostupné z: <https://www.mysql.com/>
- [4] *PostgreSQL: The World's Most Advanced Open Source Relational Database* [online]. [cit. 2021-04-13]. Dostupné z: <https://www.postgresql.org/>
- [5] *LDAP (Lightweight Directory Access Protocol)* [online]. [cit. 2021-04-09]. Dostupné z: <https://managementmania.com/cs/ldap-lightweight-directory-access-protocol>
- [6] *Optimalizace databázového výukového portálu* [online]. 2018 [cit. 2021-04-14]. Dostupné z: <https://dspace.vsb.cz/handle/10084/128610>. Diplomová.
- [7] *Angular | The modern web developer's platform* [online]. [cit. 2021-04-29]. Dostupné z: <https://angular.io/>
- [8] *Dokumentace k ASP.NET* [online]. [cit. 2021-04-29]. Dostupné z: <https://docs.microsoft.com/cs-cz/aspnet/core/?view=aspnetcore-5.0>
- [9] RANKINS, Ray, Paul BERTUCCI, Chris GALLELLI a Alex T. SILVERSTEIN. *Microsoft SQL Server 2014 Unleashed*. 1. Sams Publishing, 2015. ISBN 9780134084459.
- [10] *Dokumentace pro .NET* [online]. [cit. 2021-04-29]. Dostupné z: <https://docs.microsoft.com/cs-cz/dotnet/>
- [11] *Úvod do dotazů LINQ (C#)* [online]. [cit. 2021-04-26]. Dostupné z: <https://docs.microsoft.com/cs-cz/dotnet/csharp/programming-guide/concepts/linq/introduction-to-linq-queries>
- [12] *Dokumentace k Entity Framework* [online]. [cit. 2021-04-20]. Dostupné z: <https://docs.microsoft.com/cs-cz/ef/>
- [13] *ASP.NET MVC Pattern* [online]. [cit. 2021-04-29]. Dostupné z: <https://dotnet.microsoft.com/apps/aspnet/mvc>
- [14] *JQuery | write less, do more.* [online]. [cit. 2021-04-29]. Dostupné z: <https://jquery.com/>
- [15] *Nezpracované dotazy SQL* [online]. [cit. 2021-04-29]. Dostupné z: <https://docs.microsoft.com/cs-cz/ef/core/querying/raw-sql>

- [16] *IS administrace účtů na katedrálních databázových systémech* [online]. 2013 [cit. 2021-04-23]. Dostupné z: <https://dspace.vsb.cz/handle/10084/98993>. Bakalářská práce.
- [17] *Informační systém pro správu databázových systémů* [online]. 2015 [cit. 2021-04-12]. Dostupné z: <https://dspace.vsb.cz/handle/10084/108606>. Diplomová.
- [18] *ASP.NET | Free. Cross-platform. Open source.* [online]. [cit. 2021-04-29]. Dostupné z: <https://dotnet.microsoft.com/apps/aspnet>
- [19] *IBM Db2 | Transform and modernize your data management with AI-driven solutions* [online]. [cit. 2021-04-23]. Dostupné z: <https://www.ibm.com/analytics/db2>
- [20] *SQLite | Small. Fast. Reliable. Choose any three.* [online]. [cit. 2021-04-15]. Dostupné z: <https://www.sqlite.org/index.html>
- [21] MARTINEK, Michal. *Úvod do SQLite a příprava prostředí* [online]. [cit. 2021-04-29]. Dostupné z: <https://www.itnetwork.cz/sqlite/sqlite-tutorial-uvod-a-priprava-prostredi>
- [22] *Introduction to XML* [online]. w3schools.com [cit. 2021-04-20]. Dostupné z: [https://www.w3schools.com/xml/xml\\_what.asp](https://www.w3schools.com/xml/xml_what.asp)
- [23] *ADO.NET* [online]. [cit. 2021-04-29]. Dostupné z: <https://docs.microsoft.com/cs-cz/dotnet/framework/data/adonet/>
- [24] *What is JSON?* [online]. w3schools.com [cit. 2021-04-29]. Dostupné z: [https://www.w3schools.com/whatis/whatis\\_json.asp](https://www.w3schools.com/whatis/whatis_json.asp)
- [25] *Visual Studio | Nejlepší nástroje pro každého vývojáře* [online]. [cit. 2021-04-15]. Dostupné z: <https://visualstudio.microsoft.com/cs/>



## **A Elektronická příloha**

Elektronická příloha obsahující:

- Datová vrstva projektu DB Education
- Datová vrstva projektu DB Manager
- SQL skripty základních procedur databázových systémů pro DB Manager
- Model databáze DB Education
- Model databáze DB Manager
- Konvertor DB Education
- Konvertor DB Manager